Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 167 (2020) 1991–2002

Procedia Computer Science

www.elsevier.com/locate/procedia

International Conference on Computational Intelligence and Data Science (ICCIDS 2019)

Pattern Matching: A Quantum Oriented Approach

Kapil Kumar Soni[a]*, Akhtar Rasool[b]

[a] *Department of Computer Science and Engineering, MANIT Bhopal – 462003, INDIA*
[b] *Department of Computer Science and Engineering, MANIT Bhopal – 462003, INDIA*

Abstract

This article influences the quantum based solutions for traditional pattern matching problem, as the quantum machines can accelerate the computation speed by inherently supporting the parallel executions. The problem identifies solution by suggesting effective algorithms, and as per the classical machines several algorithms are qualified and become criterion for other, importantly noted as Knuth Morris Pratt and Boyer Moore which solves such problem in $O(M+N)$ time. A quantum identical pattern matching algorithms are based upon the principle of high speed Grover's quantum search method which searches an element over unstructured database of "N" elements in $O(\sqrt{N})$ time. The standard quantum algorithm was proposed by Ramesh and Vinay that could obtain computational speed and provides solution in $O(\sqrt{M} + \sqrt{N})$ time. We review two already existing quantum based exact and approximate pattern matching algorithms and then by combining the logic of both, a new exact pattern matching algorithm is being proposed which overcomes the algorithmic constraints and substantially proves to be equivalently better than the existing classical and quantum benchmark algorithms. So, this article includes existing and proposed quantum pattern matching algorithms, their flowcharts and examples, mathematical justification over complexity analysis. At last we discuss suitable application domains and further possible variations over the proposed work.

Keywords:Pattern Matching; Qubit; Superposition; Grover's Search; Exact and Approximate Matching.

* Corresponding author. Tel.: +91-999-345-1386.
E-mail address:prof.kapilsoni@gmail.com

10.1016/j.procs.2020.03.230

1. Introduction to Pattern Matching

The problem of pattern matching searches the existence of pattern string having length "M" = [0….M-1] within the longer text string of length "N" = [0….N-1] such that N ≥ M, sometimes the problem is to be considered as finding needle (short string) in the haystack (long string). This problem can be efficiently solved using the classical qualified pattern matching algorithms such as Knuth–Morris–Pratt (KMP), Boyer–Moore (BM) algorithm which requires at worst $O(M + N)$ pre-processing and searching time [1] or [2], but after all they became a comparative criterion for the others existing classical algorithms. This is clearly optimal, and proves its correctness on all inputs by inspecting every character of the pattern and the text i.e. kind of exact pattern matching method [3].

To begin with the introduction to quantum pattern matching algorithms, we need to understand that how such problem is being mapped over quantum. Usually, string matching solution finds all pattern occurrences in the text, such that for given pattern an algorithm outputs all start text indices "i" where T[i to i+M-1] = P[0 to M-1], $1 \leq i \leq N - M$ [3]. The same is considering with the help of Grover's quantum search which finds an index of the marked element with higher probability using amplitude amplification technique in $O(\sqrt{N})$ queries, and substantially finds better than classical approach as it requires O(N) queries to the database [6]. In Grover's search, we have an unstructured database of "N" elements along with the specific searching criteria for an element, now the algorithm tries to locate an index where the searching element must have been present with the high probability. In general, this case is reverse but somehow maps the problem of finding patterns in the quantum explored text database [7].

However, the problem may not always belong to unstructured databases, likely the databases are sometimes structured, and to get the solution through Grover's we cannot expect the efficient solution due to taking almost $O(\sqrt{N})$ time, whereas in classical approach we have two options as by applying $O(log_2 N)$ time binary searching or to satisfy the matching criteria using hash tables in O(1) time [3] or [10]. But we are intentionally finding the quantum based solutions over unstructured databases and for that the classical algorithms cannot even reach up-to the time taken by Grover's search as it leads with the quadratic polynomial speedup, so for the entire context of the article writing, we will be considering the unstructured database [4]. The Grover's search logic uses inherently a repetitive application of amplitude amplification which allows to finding marked element within $\sqrt{N}$ iterations and in each iterate it increases probability amplitude of marked element thus success probability also gets increased [5].

Indeed, the quantum pattern matching algorithms are solemnly resolved by Grover's search method, and initially Ramesh-Vinay suggested classical equivalent quantum deterministic sampling method and assures to get the solution of pattern matching problem in $O(\sqrt{M} + \sqrt{N})$ time [3]. Ramesh-Vinay algorithm takes advantage of quantum parallelism to obtain deterministic sample of pattern by pre-processing it in $\sqrt{M}$ time, whereas used logic by authors implements the search time in $O(\sqrt{N})$ time [5]. By considering Ramesh-Vinay algorithm as base, the other two quantum pattern matching algorithms were introduced without deterministic sampling and could find approximate and exact occurrence of the pattern using Grover's logic, we discuss them further in separate section.

Exact matching involves around locating all occurrences of pattern P in suitable applications like text processing, computational biology and can get classical efficient solution through Knuth-Morris-Pratt, Boyer-Moore algorithms [8] or [9]. For the processing of applications related to bioinformatics such as finding DNA, protein and amino-acid sequences within the large text, we also allow approximate matching method to find pattern within the text by allowing minimum differences up to some pre-defined threshold, and in order to find such difference for approximate matches, we use distance methods such as hamming distance, levenshtein edit distance [21] or [22].

1.1. Motivation and Contribution of Work

The quantum model of computing is emerged by providing evidence of quantum phenomenon, which leads the world towards an age of swift computation, largely extending with sufficient power to ignore classical computer incapability. Quantum computational results are probabilistic but an outcome should be deterministic, so we always look for finding solutions with high probability. Existing research contribution takes quantum computational advantage and provides an efficient solution for pattern matching problem applicable over broad area of text processing, computational biology & bioinformatics, image processing. Likewise only few quantum search based methods are available in the form of exact and approximate pattern matching that could achieve comparatively good and equivalent solutions over quantum machines. So our contribution of work is to understand the Grover's quantum search in detail, and to review existing quantum pattern matching methods with their algorithmic logic, flowcharts, and examples. We then propose quantum search based exact pattern matching method "COEQUAL Algorithm" (combines partial logics of existing quantum algorithms) which is especially designed to discover all exact occurrences of computational biology related patterns such as DNA, protein and amino acid sequences within large text. An algorithm flowchart, processing example, complexity analysis are presented with possible variations.

1.2. Organization of Work

The proposed work has been done after analyzing existing algorithms; and in fact this will eliminate the limitations of existing algorithms by suggesting new quantum based exact pattern matching algorithm called "Coequal-Algorithm". So rest of the paper is organized as per the orientation about understanding existing and proposed work methodology, as Section 2 briefly explains quantum basics, Section 3 is giving enough detailing over Grover's algorithm, apart from this we explore the existing exact and approximate quantum algorithms in Section 4. Section 5 is devoted to presenting the proposed work and at last section we conclude the work as we stated earlier.

2. Quantum Computation Basics and Properties

The quantum computation consists of a sequence of operations to be performed in parallel within the quantum state at a time. A key logic in contrast to classical computation is the processing through quantum bit i.e. qubit, a physical extension of classical bit which labels $|0\rangle = \begin{pmatrix}1\\0\end{pmatrix}$ and $|1\rangle = \begin{pmatrix}0\\1\end{pmatrix}$ as unit vectors and called as base states, in fact qubit represents two basis states $|0\rangle$ and $|1\rangle$ both be present simultaneously in superposition state "$|\psi\rangle$", where $|\psi\rangle \in \mathbb{C}^2$ in two dimensional complex vector space [10]. So the qubit $|\psi\rangle$ is a linear combination of basis state written as with computational basis "$|\psi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle$" where α_0 & α_1 are probability amplitudes with the sum of squares of both probability amplitudes (complex coefficients) equated to probability 1, i.e. to say $|\alpha_0|^2 + |\alpha_1|^2 = 1$. A state "$|\psi\rangle$" can also be represented as per the base vector "$|\psi\rangle = \alpha_0\begin{pmatrix}1\\0\end{pmatrix} + \alpha_1\begin{pmatrix}0\\1\end{pmatrix} = \begin{pmatrix}\alpha 0\\0\end{pmatrix} + \begin{pmatrix}0\\\alpha 1\end{pmatrix} = \begin{pmatrix}\alpha 0\\\alpha 1\end{pmatrix}$" [10] or [11].

Contextually, the state $|\psi\rangle$ is known as superposition state that allows the qubit to be present at once in both states. Multiple qubits remain simultaneously in superposition forms an entanglement and hence the quantum computations can support to perform exponential number of operations in just single step of execution. If a qubit $|\psi\rangle$ is measured then the outcome must be deterministic and be present either in classical state $|0\rangle$ or in state $|1\rangle$ with the probabilities $|\alpha_0|^2$ and $|\alpha_1|^2$ [10].

A quantum circuit is a composition of several quantum gates which are connected for the specific computational purpose. By default the entire quantum gated are reversible and universal in nature. These gates are used to transit a qubit value from one state to another depending on the characteristics of the quantum gate used. Quantum gates are mathematically represented as transposition matrices or linear operators and all such operators are unitary [7].

The basic gates are NOT (1-qubit) gate "X" that flips the qubit where gate operator matrix of $X = \begin{pmatrix}0 & 1\\1 & 0\end{pmatrix}$, which transforms $|0\rangle$ to $|1\rangle$ as $X|0\rangle = \begin{pmatrix}0 & 1\\1 & 0\end{pmatrix}\begin{pmatrix}1\\0\end{pmatrix} = \begin{pmatrix}0\\1\end{pmatrix}$ and $|1\rangle$ to $|0\rangle$ as $X|1\rangle = \begin{pmatrix}0 & 1\\1 & 0\end{pmatrix}\begin{pmatrix}0\\1\end{pmatrix} = \begin{pmatrix}1\\0\end{pmatrix}$. Now the important HADAMARD (1-qubit) gate "H" puts the qubit into superposition and corresponding matrix $H = \frac{1}{\sqrt{2}}\begin{pmatrix}0 & 1\\1 & 0\end{pmatrix}$, and operate over qubit to realize superposition such as $H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ and $H|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$ and equivalently written as $H|i\rangle = \frac{1}{\sqrt{2}}(|0\rangle + (-1)^i|1\rangle)$ [10], and further for "n" qubits system $H^{\otimes n}|0\rangle^{\otimes n} = \frac{1}{\sqrt{2^n}}\sum_{i=0}^{2^n-1}\alpha_i|i\rangle$, where $N = 2^n$. Two more gate such as CNOT (2-qubits) and CCNOT (3-qubits) gates are used for flipping the target qubit "q_t" on the basis of high enabled control qubit(s) "q_c", basically CNOT implements in Grover's oracle that flips the sign of ancilla (target) qubit if control qubit sets to $|1\rangle$ by doing XOR, therefore CNOT $\begin{pmatrix}\mathrm{qc}\\\mathrm{qt}\end{pmatrix} = \begin{pmatrix}\mathrm{qc}\\\mathrm{qc} \oplus \mathrm{qt}\end{pmatrix}$, CCNOT classically equivalent to AND, and flips target if inputs set to $|1\rangle\,|1\rangle$ and realizes as $\begin{pmatrix}\mathrm{qc1}\\\mathrm{qc2}\\\mathrm{qt}\end{pmatrix} = \begin{pmatrix}\mathrm{qc1}\\\mathrm{qc2}\\\mathrm{qt} \oplus (\mathrm{qc1}\,.\,\mathrm{qc2})\end{pmatrix}$ [10].

3. Pattern Matching Background: Grover's Quantum Search Method

A Grover's quantum search method finds the given marked element occurrence in randomly ordered database of "N" elements in "$\sqrt{N}$" queries, and in contrast classical method requires "N", and other than Grover's no other quantum solvable method is available. The search problem have solution, suggested by Grover's in the form of quantum algorithm and its equivalent circuits which reports the probabilistic occurrence of such element in O($\sqrt{N}$ $\log_2 N$) times (logarithmic factor considered for "n" qubits Hadamard operations as n = $\log_2 N$). So, Grover's asymptotically justified as an optimal algorithm that determines the search string through function evaluates to true.

The quantum search algorithm runs on amplitude amplification logic which allows finding marked element within $\sqrt{N}$ iterations, and each iteration increases complex coefficients or probability amplitude of marked element by constant factor and thus success probability also gets increased. The probability amplitudes are weights associated with qubits present in superposition, and during measurement the square of coefficients generate the high success probability of obtaining solution. Grover's algorithm explores the concept of amplitude amplification and hits the target in quadratic polynomial time, thus achieves speed-up over classical algorithm [6] or [7].

The number of iterations or rotations is bounded by $\sqrt{N}$ because super-positioned state $|\psi\rangle$ needs to be shifted toward the solution state over two dimensional plane. A superposition state makes an angle $(^{\pi}/_{2} - \theta)$ and each Grover's rotation needs 2θ. Therefore to get the upper bound over "R" iterations we need $(R * 2\theta = {}^{\pi}/_{2} - \theta)$ then "R" evaluates to $(R = {}^{\pi}/_{4\theta} - {}^{1}/_{2})$ [14],[15]. As depicted in Fig. 1(a), we show how the superposition state of an element is being transformed into solution state. The superposition of all elements $\frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} \alpha_i\ |i\rangle$ where N = 2^n, so one solution be available in the form of probability amplitude out of ${}^{1}/_{\sqrt{N}}$ elements, and hence solution state is perpendicular to base state, which makes sine angle as $\sin\theta = {}^{1}/_{\sqrt{N}}$ and $\theta = {}^{\pi}/_{2}$. And for large "N", $\sin\theta = \theta$, now substituting θ in "R" value, then $(R = {}^{\pi}/_{4}\sqrt{N} - {}^{1}/_{2})$ [14] or [15] or [16].

Grover's quantum circuit takes "n" qubits quantum index register and an oracle qubit; both are initialized to $|0\rangle^{\otimes n}$ and $|0\rangle$, and we pass searching criteria to workspace qubits which functionally matches at $|i\rangle$ text index. Now, put all the index qubits into superposition state through Hadamard gate as $|\psi_1\rangle = \sum_{i=0}^{2^n-1} \alpha_i\ |i\rangle$ with $\sum_{i=0}^{2^n-1} |\alpha_i|^2 = 1$ and oracle to $|1\rangle$ then put oracle into superposition. Theoretically we require $\frac{\pi}{4}\sqrt{N}$ rotations of Grover's operator and in each iteration oracle marks search element and then diffusion operator amplifies the amplitude of marked element state by doing inversion about amplitude mean, now for pictorial circuit description of Grover's search refer Fig. 1(b). The oracle does phase flip of marked element at index $|i\rangle = (-1)^{f(|i\rangle)}\ |i\rangle$, and diffusion operator circuit does the "$H^{\otimes n}$ $(2|0\rangle\langle 0| - I)$ $H^{\otimes n}$ = $(2|\psi\rangle\langle\psi| - I)$" [20], it means initially compute the average of all probability amplitudes saying $\alpha \leftarrow \frac{1}{N} \sum_i \alpha_i$ and then recalculate the new probability amplitudes for obtaining superposition state $|\psi_2\rangle$ and such calculation is done by $\sum_{i=0}^{N-1} (2\alpha - \alpha_i)\ |i\rangle$. At last we measure the "n" qubits index register for obtaining solution at index $|i\rangle$ with high probability [16] or [19].

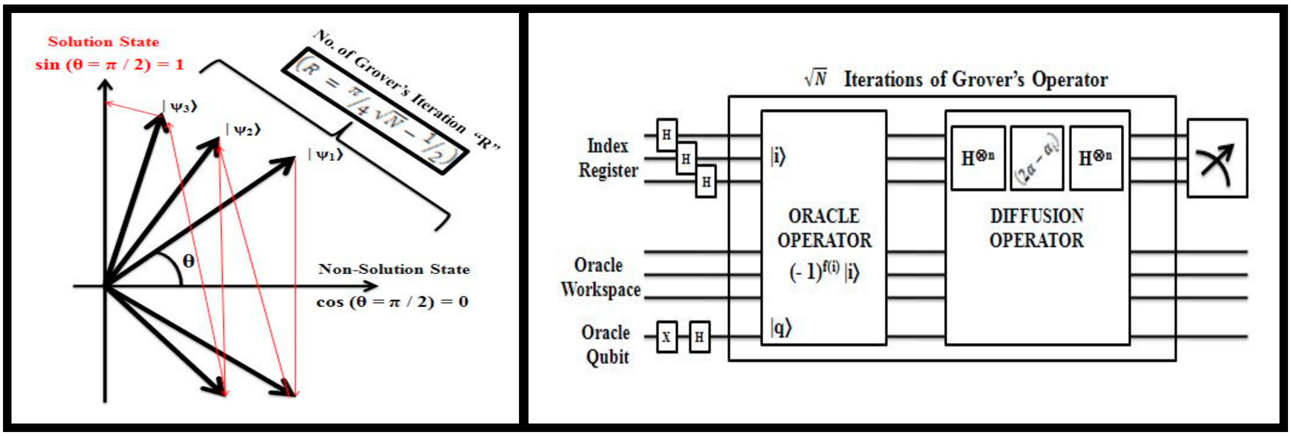


Fig. 1: **(a)** Geometric Representation for obtaining Solution State; **(b)** Pictorial Representation of Grover's Circuit

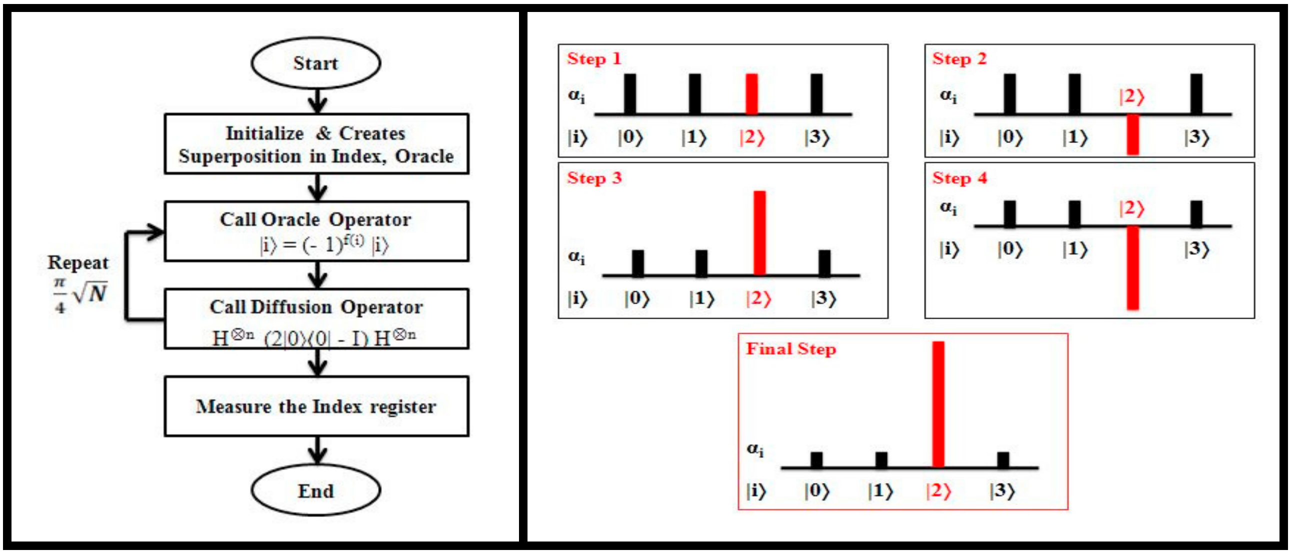


Fig. 2: **(a)** Grover's Algorithm Flowchart; **(b)** Grover's Search Example

The algorithmic steps are shown through the flowchart and Grover's example is depicted in Fig. 2(a) and 2(b). In example out of four elements we intend to search an element situated at index$|2\rangle$. The actual computational logics are written in ALGO – 1, we only need to notice about the iterative step of algorithm, where oracle is implemented by CNOT gate just to negate the amplitude of marked element whereas diffusion operator is series combination of Hadamard, CNOT^{n-1} and again Hadamard gates that implements inversion about the mean [10].

ALGO – 1: ***Grover's Quantum Search Algorithm***

Input: Index *Register "**i**", Oracle qubit "**q**", Workspace qubits "**w**"*

Initialization Step: $|i\rangle^{\otimes n} \leftarrow |0\rangle^{\otimes n}$ **and** $|q\rangle \leftarrow |0\rangle$

Superposition Step: $H^{\otimes n}\,|i\rangle^{\otimes n} \rightarrow |\psi\rangle = \frac{1}{\sqrt{2^n}}\sum_{i=0}^{2^n-1}\alpha_i|i\rangle$

$X|q\rangle = |1\rangle \rightarrow H|1\rangle \rightarrow \frac{|0\rangle+|1\rangle}{\sqrt{2}}$

Iteration Step : **Repeat for** $\frac{\pi}{4}\sqrt{N}$ **times**

Apply Oracle Operator "*O*"

$O\,(|i\rangle\,|q\rangle) \rightarrow (-1)^{f(|i\rangle)}\,|i\rangle\,|q\rangle)$

// "O" flips $|i\rangle$ to $-|i\rangle$ if $|i\rangle$ = = w, $|q\rangle$ unchanged

Apply Diffusion Operator "*D*"

$(2|\psi\rangle\langle\psi| - I)\sum_i \alpha_i = \sum_{i=0}^{2^n-1}(2\alpha - \alpha_i)\,|i\rangle$

// "D" does inversion about the mean $\alpha \leftarrow \frac{1}{N}\sum_i \alpha_i$

Measurement Step: **Measure Index Register to obtain Solution State with High Probability "*p*" ≥ $^3/_4$.**

Element Occurrence(s) Theorem: Given "N" elements with "t" marked items, an oracle returns one random element out of "t ≥ 1" that satisfies the search criteria in $\sqrt{\frac{N}{t}}$ query complexity as well O($\sqrt{\frac{N}{t}}\log_2 N$) time complexity and with probability "*p*" ≥ $^3/_4$. So, if for "t = N" algorithm requires to find all elements occurrences "t" as t * $\sqrt{\frac{N}{t}} = \sqrt{t}\,\sqrt{t}\,\sqrt{\frac{N}{t}} = \sqrt{Nt}$ query and O($\sqrt{Nt}\log_2 N$) time complexity [3] or [5] or [17].

Proof: As stated earlier, that theoretically we require $\sqrt{N}$ iterations to reach solution states from superposition, and we get one solution. But, here we notice that oracle marks all such elements that satisfies searching criteria in constant time, so to proceed for obtaining single specific solution we need amplitude amplification and both sub-steps are required to be executed for $\sqrt{N}$ iterations, hence concludes Theorem 1, as best to find all occurrences.

Analysis Note – 1: The algorithm is based on time required for performing operations at Grover's iteration step. Do remember that we need $\log_2 N$qubits due to processing entire search space, and all of them are processed in superposition through Hadamard gate operations, and Hadamard takes constant time, so for $\sqrt{N}$ iterations the concatenation of Hadamard related operations are done over $\log_2 N$ qubits, and hence for entire computations the total time required by the algorithm is O($\sqrt{N}\log_2 N$).

4. Existing Quantum Approximate and Exact Pattern Matching Algorithms

The Grover's search method extends to find the pattern matching solution in the large text, we know that the competent solutions exist but the available quantum based solutions are expected as it can achieve computational speedups. This section explores the algorithmic logic and illustrations of approximate and exact quantum pattern matching algorithms followed by complexity analysis. The common objective of the algorithm is to determine pattern "P" of length "M" in given text "T" of length "N" within the quantum computational time [3] or [18].

4.1. Quantum Approximate Pattern Matching Algorithm

Definition: The Pattern "P" having length "M = [0….M – 1]" and Text "T" of length "N = [0….N – 1]" where both P, T ∈ Σ and N ≥ M, with Hamming Distance "H" calculates at index "|i⟩" of text such that H(T[i, ... i+M], P]) finds ≤ "d" ∈ Distance(T, P) with high probability.

An approximate matching is having wide application areas, and this algorithm suggests the method of finding such approximate pattern occurrences in the large text. Inherently the algorithmic logic uses Grover's search technique for obtaining solution speedup. The algorithmic logic is based on the working of two phases or sub-part of the algorithm such as Filtering and Verification. And for better approximation between text substring and pattern this uses Hamming Distance "H" method that identifies possible character mismatches from index "|i⟩" as per aligned H(T[i, i+M-1], P) = $\sum_{j=0}^{M-1} g(T[i+j], P[j])$ where g(a, b) = 1, if a ≠ b, otherwise 0. The pattern occurrence at index "|i⟩" is only reported if the H(T[i, i+M-1], P) ≤ "d" i.e. Threshold Distance "d", with high probability [12].

The Fig. 3(a) shows algorithmic flowchart and 3(b) depicts the algorithm steps with the help of example, in which the text indices are filtered as per the possible start of the pattern where hamming distance "H" is found less than the threshold distance "d" and other indices are discarded due to the trimming logic of search space. Now, the identified indexes are passed to the pattern verification phase which inherently utilizes the Grover's search logic to obtain the pattern occurrence at index "|i⟩" with high probability.

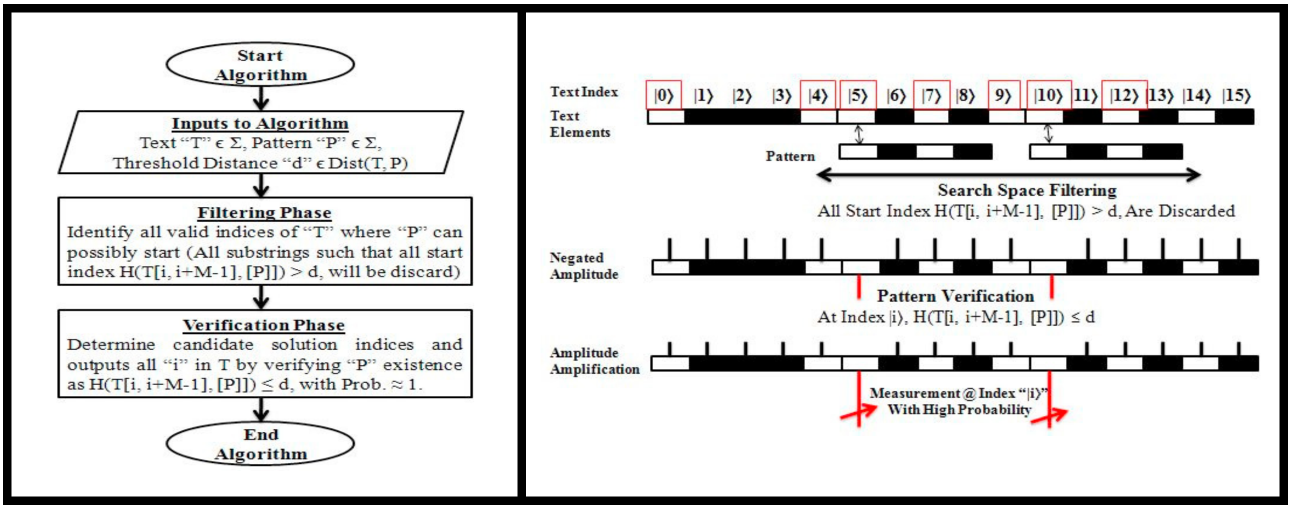


Fig. 3: **(a)** Approximate Algorithms Flowchart; **(b)** Approximate Algorithm Example

The detailed algorithm is written in ALGO – 2, and the pattern matching process is based on prior filtering of search space and then verification at identified indices. As per the algorithm, we show that filtering identifies all indices $|i\rangle$ where pattern starts in text. FILTERING initializes the text in index register $|R_I\rangle$, and then creates the superposition in it, now marks first symbol $|i_j\rangle$ as start of pattern, and then finds pattern locations in $|R_I\rangle$ and lastly copies $|i\rangle$ indices from $|R_I\rangle$ to $|R_L\rangle$. Now, VERIFICATION identifies mismatch between pattern and text substring at index $|i\rangle \in |R_L\rangle$, now, computes hamming distance at mismatch register, then compare it with threshold value and if $H \leq d$, then mark valid indices and keep them in $|R_I\rangle$. Finally do the amplitude amplification at index $|i\rangle$ and after desired Grover's rotation, measure the index register to report "P" in "T" at $|i\rangle$ [12].

ALGO – 2: ***Quantum Approximate Pattern Matching Algorithm***

Input: *Text "T" Length "$N = 2^{n=qubits}$", Pattern "P" Length "M", $N \geq M$*
Threshold "d", First Symbol "i_j", Pattern Loc"$i - i_j$", Oracle Qubit "q"
Dedicated Quantum Registers For Filtering & Verification Phases

FILTERING_ALGORITHM

Initialize Text: $|R_I\rangle \leftarrow |i\rangle^{\otimes n} = |0\rangle^{\otimes n}$ and $|q\rangle \leftarrow |0\rangle$ // $0 \leq |i\rangle \leq N\text{-}1$

Create Superposition: $|R_I\rangle \leftarrow |\psi\rangle = \frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} \alpha_i |i\rangle\, |0\rangle$

Mark First Symbol: $|R_I\rangle = \frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} \alpha_i |i\rangle\, |i_j\rangle$

Find "P" Location: $|R_I\rangle = \frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} \alpha_i |i\rangle\, |i - i_j\rangle$

Fix "P" Location: $|R_L\rangle \leftarrow |R_I\rangle = (|i\rangle\, |q\rangle \rightarrow (-1)^{f(|i\rangle)}\, |i\rangle\, |q\rangle)$
// For all $|i\rangle \in |i - i_j\rangle$, we get $-|i\rangle$, if $T[i, i+M-1] = P$

VERIFICATION_ALGORITHM

Identify Mismatch: $|R_{MR}\rangle = \frac{1}{\sqrt{2^n}} \sum_{i \in R_L} |i\rangle\, |T[i, i+M-1]\rangle\, |P\rangle\, |i_{mis}\rangle$

Compute "H" Distance: $|R_H\rangle = \frac{1}{\sqrt{2^n}} \sum_{i \in R_L} |i\rangle\, |T[i, i+M-1]\rangle\, |P\rangle\, |i_{MR}\rangle\, |i_H\rangle$

Marking Valid Indices: $|R_I\rangle \leftarrow |R_{VIR}\rangle = \frac{1}{\sqrt{2^n}} \sum_{i \in R_L} |i\rangle\, |T[i, i+M-1]\rangle\, |P\rangle\, |i_{MR}\rangle\, |i_H\rangle\, |d\rangle$

Amplitude Amplification: $|R_I\rangle = (2|\psi\rangle\langle\psi| - I) \sum_{i \in R_I} |i\rangle = \sum_{i=0}^{2^n-1} (2\alpha - \alpha_i)\, |i\rangle$

Measure Index: **Measure Index Register R_I to report pattern "P" at index $|i\rangle$ with high probability.**

Analysis Note – 2: This algorithm finds multiple occurrences of the same pattern but limited to process only single pattern, not working for finding multiple patterns in the large text. A better approximation approach can be found by implementing the other edit distance calculation logic than the hamming distance. The FILTERING requires $O(N \log_2 N)$ time for marking the possible start location of the pattern and to trimming out the entire search space then VERIFICATION algorithm outputs all indices "$|i\rangle$" where pattern occurrence is reported, such that $H(T[i, i+M-1], P) \leq d$ with the time complexity $O(\sqrt{N} \log_2 N)$ and with high probability [12].

4.2. Quantum Exact Pattern Matching Algorithm

Definition: The Pattern "P" with length "M = [0….M – 1]" and Text "T" of length "N = [0….N – 1]" where both $P, T \in \Sigma$ and $N \geq M$, with parallel match sliding window that finds exact match of "P" in "T" if oracle function f(i) evaluates as f(i) = 1, (if T[i+1 to i+M-1] = P[0 to M-1]) with high probability.

As the quantum computational model is probabilistic, so in order to find the pattern occurrences in the text, an exact matching logic may be comparatively more suitable to the desired applications. This algorithm marks the solution by "Sliding Window" concept and find shift over index register in order to report the occurrence of pattern. Sliding window is based upon superposition property that matches the aligned text substring with pattern in single execution step. Once this shift is found, then the algorithm inherently utilizes Grover's search logic for validating the pattern occurrence at index "|i⟩", and then algorithm does the measurement and verification of solution with high probability [13].

ALGO – 3: *Quantum Exact Pattern Matching Algorithm*

Input: *Text "**T**" Length "**N** = $\mathbf{2^{n=qubits}}$", Pattern "**P**" Length "**M**", **N ≥ M***
Dedicated Quantum Registers For Text & Pattern Processing

Initialize Text: $|R_I\rangle \leftarrow |i\rangle^{\otimes n} = |0\rangle^{\otimes n}$ and $|q\rangle \leftarrow |0\rangle$ // $0 \leq |i\rangle \leq N-1$

Create Superposition: $|R_I\rangle \leftarrow |\psi\rangle = \frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} \alpha_i |i\rangle$

$X|q\rangle = |1\rangle \rightarrow H|1\rangle \rightarrow \frac{|0\rangle + |1\rangle}{\sqrt{2}}$

Mark Solution Index: $f(i) = \begin{cases} 0, & if\ T[i\ to\ i+M-1] \neq P[0\ to\ M-1] \\ 1, & if\ T[i\ to\ i+M-1] = P[0\ to\ M-1] \end{cases}$

Grover's Iteration: **Repeat for $\frac{\pi}{4}\sqrt{N}$ times**

Apply Oracle Operator "*O*"

$O\,(|i\rangle\,|q\rangle) \rightarrow (-1)^{f(|i\rangle)}\,|i\rangle\,|q\rangle)$

// "O" flips |i⟩ to –|i⟩ if T[i, i+M-1] = P, |q⟩ unchanged

Apply Diffusion Operator "*D*"

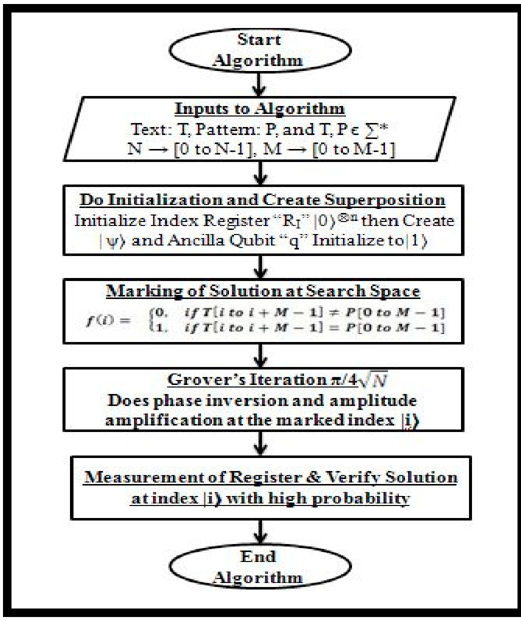
$(2|\psi\rangle\langle\psi| - I) \sum_i \alpha_i = \sum_{i=0}^{2^n-1} (2\alpha - \alpha_i)\,|i\rangle$

// "D" does inversion about the mean $\alpha \leftarrow \frac{1}{N}\sum_i \alpha_i$

Measure Index: **Measure Index Register R_I to report pattern "P" at index |i⟩ with high probability.**

Verify Solution: **Verify Solution $T[i\ to\ i+M-1] = P[0\ to\ M-1]$ with |i⟩ as valid shift value.**

We explore algorithm steps and exact pattern matching example in Fig. 4(a) and 4(b), in which sliding window is aligned each next index from left to right and checks for the output of f(i) = 1, if it happens then the corresponding index "|i⟩" is selected for Grover's iteration to negate the phase and then for doing amplitude amplification. Lastly we can measure the occurrence of the pattern at index "|i⟩" with high probability.



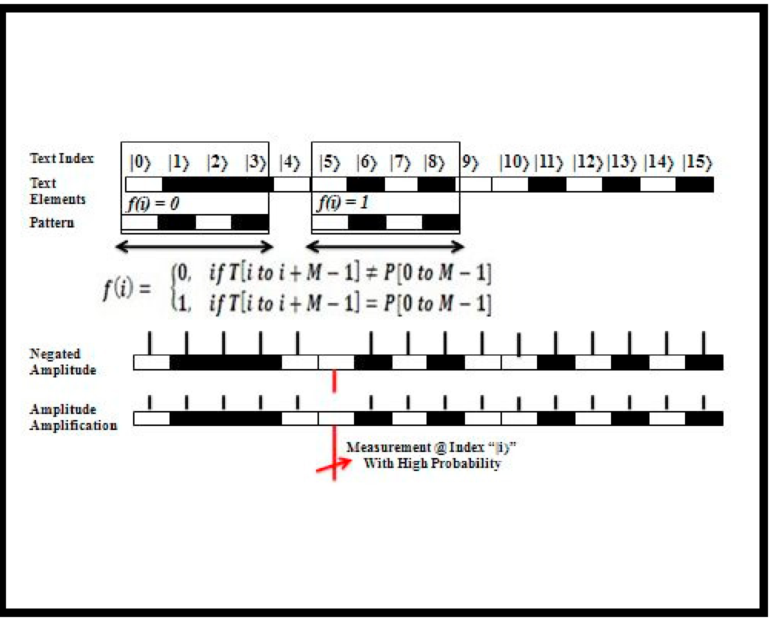


Fig. 4: **(a)** Exact Algorithms Flowchart; **(b)** Exact Algorithm Example

Analysis Note – 3: The algorithm is limited to find the leftmost / single occurrence of pattern and only suggested for processing single pattern. The complexity of entire algorithm is depending on step 4, which need to be iterated O($\frac{\pi}{4}\sqrt{N}$) or O($\frac{\pi}{4}\sqrt{2^n}$) time, so in general the complexity of algorithm can be considered exponential in number of qubits related operations but still linear in O($\sqrt{N}$) time. As noted that the actual time required for such Hadamard related computations are $\log_2 N$. So the concluding complexity of an algorithm takes $O\left(\sqrt{N}\ \log_2 N\right)$ time.

4.3. Constraints and Limitations of Approximate and Exact Pattern Matching Algorithms

Approximate matching method requires better approximation approach to find solution indices in relatively less edit distance and with more accuracy. The idea can be expanded for exact pattern matching technique. Exact matching technique limits to find leftmost occurrence of the pattern, so same sliding window concept generalization is possible for finding multiple occurrences of same pattern. An equivalent approximation based approach can take the advantage of algorithm and be more suitable to the application areas [12] or [13].

5. Proposed Quantum Pattern Matching Algorithm: **CO**mbined **E**xact **QU**antum **AL**gorithm **(COEQUAL)**

We expect exact matching problem to be more realistic in the field of biological science, other than this approximate matches are still meaningful. Biologists investigate nucleotide or amino acid types of pattern in the large sequence biological databases. Exact string matching algorithms are essential components in DNA analysis of the computational biology. The equivalent quantum exact matching method is constrained to only finding left most occurrence of the pattern, whereas this is usually impractical, due to the database sequences may have large number of pattern occurrences and an algorithm should be capable of reporting them. So, to overcome the existing algorithm limitation, we propose the new quantum exact matching algorithm that partially combines existing approximate and exact algorithms and thus we call "COEQUAL Algorithm".

5.1. COEQUAL's Logic Design and Processing Methodology

As existing method was limited to find single occurrence of pattern, we used search space filtering method of approximate algorithm to mark the possible locations of pattern in the text index register R_I, and then we store the marked locations in location register R_L, thus we gave the computational processing logic through which sliding window method could be applied to find the pattern occurrences at the identified indices. The proposed processing methodology is demonstrated with example mentioned in Fig. 5, and algorithmic working is explored in flowchart of Fig. 6, both pictures are recommended before proceeding to the COEQUAL Algorithm.

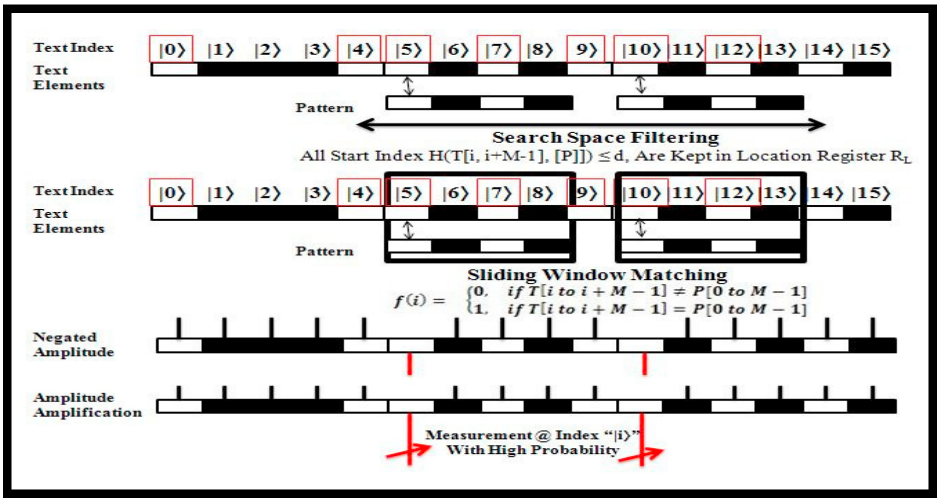


Fig.5: Example based Demonstration of COEQUAL Algorithm Processing

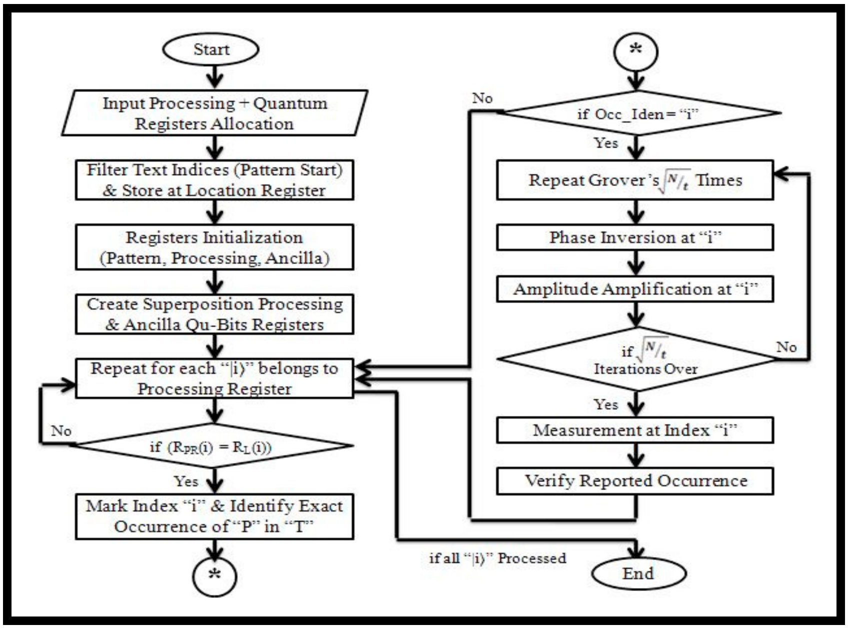


Fig.6: Flowchart of COEQUAL Algorithm

5.2. COEQUAL Exact Pattern Matching Algorithm

PROPOSED ALGO: ***COEQUAL Algorithm***

Input: *Text "**T**" Length "$N = 2^{n=qubits}$", Pattern "**P**" Length "**M**", $N \geq M$*
Dedicated Quantum Registers For Text Filtering, Text & Pattern Processing

Text Filtering: $|R_L\rangle \leftarrow$ **FILTERING_ALGORITHM**$(|R_I\rangle)$;

Initialize Text: $|R_{PR}\rangle \leftarrow |i\rangle^{\otimes n} = |0\rangle^{\otimes n}$ **and** $|q\rangle \leftarrow |0\rangle$

Create Superposition: $|R_{PR}\rangle \leftarrow |\psi\rangle = \frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} \alpha_i |i\rangle$

$X|q\rangle = |1\rangle \rightarrow H|1\rangle \rightarrow \frac{|0\rangle + |1\rangle}{\sqrt{2}}$

COEQUAL Logic: ***Repeat for each*** $|R_{PR}\rangle|i\rangle$ *// $0 \leq |i\rangle \leq N\text{-}1$*

{ ***if*** $(|R_{PR}\rangle|i\rangle == |R_L\rangle|i\rangle)$ ***then***

{ *// Mark Solution Index*

$$f(i) = \begin{cases} 0, & if\ T[i\ to\ i+M-1] \neq P[0\ to\ M-1] \\ 1, & if\ T[i\ to\ i+M-1] = P[0\ to\ M-1] \end{cases}$$

if $(|R_{PR}\rangle T[|i\rangle\ to |i\rangle+M\text{-}1] == P[0\ to\ M\text{-}1]$

{ *// Grover's Iteration*

Repeat for $\frac{\pi}{4}\sqrt{N}$ **times**

{ $O\ (|i\rangle\,|q\rangle \rightarrow (-1)^{f(|i\rangle)}\,|i\rangle\,|q\rangle)$

$(2|\psi\rangle\langle\psi| - I) \sum_i \alpha_i = \sum_{i=0}^{2^n-1} (2\alpha - \alpha_i)\,|i\rangle$

}

Measure Processing Register R_{PR} ***at*** $|i\rangle$;

Verify Solution $T[i\ to\ i+M-1] = P[0\ to\ M-1]$;

} *// End of Inner if*

} *// End of Outer if*

} *// End of Outer Repeat*

In the initial step of algorithm, we call FILTERING_ALGORITHM by passing the index register R_I, and get resultants in terms of possible start indices of pattern "P" in text "T", and all such filtered indices are then be stored in location register R_L. Now, algorithm takes processing register R_{PR}, which is intentionally taken to process the entire search space with all possible text indices $0 \le |i\rangle \le N-1$, and an initialization along with transforming it into superposition is required, in addition to this we consider ancilla qubit to process inside Grover's operator. After all COEQUAL's processing logic begins that explore R_{PR} for each random index $|i\rangle$ and then perform check with R_L, if compared indices are found same, so we use sliding window method to parallel check for an exact occurrence of the pattern at $|i\rangle$. In the event of matched index, we call Grover's which iterates $\sqrt{N}$ times and does amplitude inversion and inversion about mean to amplify the probability amplitude of marked pattern index. However, after completion of successful repetition of Grover's we can be able to report index position $|i\rangle$ as valid shift and verify the pattern occurrence as T[i to i+M-1] = P[0 to M-1] with high probability.

5.3. COEQUAL's Algorithm Justifications and Complexity Discussion

The algorithm COEQUAL combines partial processing logics of existing algorithm, and provides a method to find all pattern occurrences within the larger texts, thus overcomes the constraint of existing method. The algorithm has been justified by suggesting the example, flowchart along with the processing steps defined inside COEQUAL Algorithm. Now, if we have a look over complexity of the algorithm, then this must be noted in defined context of FILTERING_ALGORITHM which takes $O(N \log_2 N)$ time for marking the possible start location of the pattern and to trimming out the entire search space (Refer Analysis Note – 2), and COEQUAL Logic that does parallel match through sliding window in O(1) time, followed by repeated application of Grover's operator $\sqrt{N}$ times, but as noted earlier that the actual time required for Hadamard related computations are $\log_2 N$ hence takes $O(\sqrt{N} \log_2 N)$ time (Refer Analysis Note – 3). So, in $O(\sqrt{N} \log_2 N)$ it searches for one pattern, now to search for all "t" occurrences of "P", we consider Element Occurrence(s) Theorem (Refer Section – 3), and as per the theorem we require $\sqrt{Nt}$ query and $O(\sqrt{Nt} \log_2 N)$ time complexity. Therefore, total time required to find all patterns in text –

$$\textbf{TIME (COEQUAL Algorithm)} = \boldsymbol{O(N \log_2 N) + O(\sqrt{Nt}\, log_2 N)}$$

5.4. COEQUAL's Performance, Applications and Future Works

As per the time complexity justified over COEQUAL Algorithm, we consider the case where text "T" contains large number of pattern occurrences "t" which is supposed to be equal to "N" i.e. ($t \approx N$), then in worst our COEQUAL Algorithm performs equivalently to the search time required by classical benchmark algorithm such as KMP O(N) and quantum competent Ramesh and Vinay $O(\sqrt{Nt})$ as = $O(\sqrt{NN} \log_2 N) = O(\sqrt{N^2} \log_2 N) = O(N \log_2 N)$. Remember that the logarithmic factor can be ignored here, just because it is required for Hadamard related quantum circuit operations, need not to be considered explicitly.

For the processing of applications related to various domains of computer science namely information retrieval, and very specific computational biology such as finding DNA, protein and amino-acid sequences within the large text, so exact pattern matching algorithms will be more fruitful. In future, the algorithm looks for better pre-processing of search space, and the algorithm design can also be extended for processing multiple patterns within the large text, as multiple pattern matching have more practicability. As Grover's have possible variation that allows dynamic search of element, so similarly the exact matching algorithm can develop for processing dynamic patterns.

6. The Conclusion

Mainly research is oriented about exploring and suggesting the quantum based pattern matching solutions which either improves computational complexity or provides competent solution equivalent to classical pattern matching algorithms. In purview of quantum algorithms, we analyzed the fact that such algorithms were implicitly utilizing the Grover's quantum searching logic as it could achieve computational speedup. We focused on Grover's over complexity analysis and elements occurrences findings. The two existing quantum exact and approximate pattern matching were discussed in detail and we laid down their computational limitations. We intended towards the new design of exact pattern matching algorithm that partially combines existing methods and proposed logic through which multiple occurrences of same pattern could be reported. The proposed COEQUAL Algorithm theoretically proved to be the competent solution for exact quantum pattern matching than others classical or quantum algorithms. COEQUAL Algorithm will work equivalently better than classical counterparts as it justifies for finding pattern in perspective of large number of occurrences.

Acknowledgements

The authors are thankful for other researchers working in the same domain, and sharing their ideas to be extended over upcoming quantum computing technology. We also welcome the suggestions from the reviewers so the quality of article can be improved.

References

[1] Knuth, D. E, Morris JR J. H and Pratt V. R. (1977) “Fast pattern matching in strings” SIAM Journal Computing 6 (1): 323–350.

[2] Boyer, R. S. and Moore, J. S. (1977) “A fast string searching algorithm” Communication of ACM 20 (10): 762–772.

[3] H. Ramesh & V. Vinay. (2003) “String Matching in O($\sqrt{n}+\sqrt{m}$) quantum time”, Elsevier Journal of Discrete Algorithms 1 (2): 103–110.

[4] Paulo Mateus. (2003) “A Quantum Algorithm for Closest Pattern Matching” International Journal of Theoretical Physics 52 (11): 3970–3980.

[5] Chris Lomont. (2003) “Robust String Matching in O($\sqrt{N}$ + M) Quantum Queries”, Mathematics Subject Classification 2003.

[6] Lov K. Grover. (1996) “A fast quantum mechanical algorithm for database search” In: Proceeding of ACM STOC, 212–219, May 1996.

[7] Akhtar Rasool and Kapil Kumar Soni. (2018) “Searching Speedup through Quantum Computing” Journal of Advance Research in Dynamical & Control Systems (JARDCS) 10 (15): 113–123.

[8] Nimisha Singh, Deepak Garg. (2012) “String Matching Algorithms and their Applicability in various Applications” International Journal of Soft Computing and Engineering 1 (6): 156–161.

[9] Kapil Kumar Soni. (2016) “Contributing Efforts of Various String Matching Methodologies in Real World Applications” International Journal of Computer Science and Engineering 4 (1): 82–85.

[10] Michael A. Nielsen and Isaac L. Chuang. (2010) “Quantum Computation and Quantum Information”, 10th Anniversary Edition, 2010, UK: Cambridge University Press.

[11] Jasmeet Singh and Mohit Singh. (2016) “Evolution in Quantum Computing” In: IEEE 5th International Conference on System Modelling & Advancement in Research Trends (SMART) 2016: 267–270.

[12] Jeffrey Aborot. (2017) “Quantum Approximate String Matching for Large Alphabets” Theory & Practice of Computation, Proceedings of workshop on computation (WSPC) World Scientific 20 (8): 37–50.

[13] De Jesus B.K.A., Aborot J.A., Adorna H.N. (2013) “Solving the Exact Pattern Matching Problem Constrained to Single Occurrence of Pattern P in String S Using Grover’s Quantum Search Algorithm” In: Theory & Practice of Computation, Proceedings in Information & Communications Technology (WCTP), Springer Tokyo PICT 7: 124–142.

[14] Aamir Mandviwalla, Keita Ohshiro and Bo Ji. (2018) “Implementing Grover’s Algorithm on the IBM Quantum Computers” In: 2018 IEEE International Conference on Big Data (Big Data) 2018: 2531–2537.

[15] Gilles Brassard, Peter Hoyer Michele Mosca and Alain Tapp. (2002) “Quantum Amplitude Amplification and Estimation” Quantum Computation and Quantum Information, AMS Contemporary Mathematics, 305 (1): 53–74.

[16] Michel Boyer, Gilles Brassard, Peter Hoyer and Alain Tap. (1998) “Tight Bounds on Quantum Searching” Quantum Physics 4 (5): 493–505.

[17] Naphan Benchasattabuse, Prabhas Chongstitvatana and Chatchawit Aporntewan. (2018) “Quantum Rough Counting and Its Application to Grover’s Search Algorithm”, In: IEEE 3rd International Conference on Computer and Communication Systems (ICCCS) 2018: 21–24.

[18] Ashley Montanaro. (2017) “Quantum Pattern Matching Fast on Average” Springer Journal Algorithmica, 77 (1): 16–39.

[19] Kai Kang. (2015) “Two Improvements in Grover’s Algorithm” In: IEEE 27th Chinese Control and Decision Conference 2015: 1179–1182.

[20] Indranil Chakrabarty, Shahzor Khan and Vanshdeep Singh. (2017) “Dynamic Grover Search: Application in Recommendation System and Optimization Problems” Quantum Information Process (QIP) 16 (153): 152–172.

[21] Eric Rivals, Leena Salmela, Jorma Tarhio. (2011) “Exact Search Algorithms for Biological Sequences” Algorithms in Computational Molecular Biology: Techniques, Approaches and Applications, John Wiley & Sons 2011: 91–111.

[22] Halil Nusret Bulus, Aydin Carus, Altan Mesut. (2017) “A new word-based compression model allowing compressed pattern matching” Turkish Journal of Electrical Engineering & Computer Science, 25 (1): 3607–3622.